UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/043,223	01/14/2002	Gil Vinitzky	1102-US	6216

7590 12/19/2005

EDWARD LANGER, PAT. ATTY.
C/O SHIBOLETH, YISTAELI,ROBERTS, ZISMAN& CO.
350 FIFTH AVENUE, 60TH FLOOR
EMPIRE STATE BUILDING
NEW YORK, NY 10118

EXAMINER	
RIZZUTO, KEVIN P	

ART UNIT	PAPER NUMBER
2183	

DATE MAILED: 12/19/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/043,223	VINITZKY, GIL	
	Examiner	Art Unit	
	Kevin P. Rizzuto	2183	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *27 September 2005*.

2a)☒ This action is **FINAL**. 2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-4 and 6-23* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-4, 6-23* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☐ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____.
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

1. Claims 1-4 and 6-23 have been examined.

Papers Submitted

2. It is hereby acknowledged that the following papers have been received and placed on record in the file: Amendment on 9/27/2005.

Specification

3. The lengthy specification has not been checked to the extent necessary to determine the presence of all possible minor errors. Applicant's cooperation is requested in correcting any errors of which applicant may become aware in the specification.

Withdrawn Objections/Rejections

4. Applicant, via amendment, has overcome the objection to the title set forth in the previous Office Action. Consequently, this objection has been withdrawn by the Examiner.

5. Applicant, via amendment, has overcome the 35 U.S.C. 112 Rejections to claims 7-11 set forth in the previous Office Action. Consequently, these Rejections have been withdrawn by the Examiner.

Claim Rejections - 35 USC § 112

6. The following is a quotation of the second paragraph of 35 U.S.C. 112:

> **The specification shall conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention.**

7. Claims 1-4 and 6-23 are rejected under 35 U.S.C. 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter which applicant regards as the invention.

8. As per claim 1, the claim limitations include, "n virtual multithreaded processors...dividing each phase of said k-phased pipeline of said computer processor configuration into a plurality of n sub-phases...creating n virtual pipelines within said k-phases pipeline, wherein each of said n virtual pipelines comprises n*k sub-phases:" The relationship between the variable "n", the "virtual pipelines" and the "virtual multithreaded processors" is unclear and renders the claim indefinite. It is unclear whether the "n virtual multithreaded processors" are the same entities as the "n virtual pipelines", if the "n virtual multithreaded processors" are different than the "n virtual pipelines", and if they are different entities, if it is coincidental that there is the same number of each or if there is a specific purpose to this (one is not found in the specification). Evidence in the specification and claims can be found pointing to both interpretations. For instance, there are "n*k" sub-phases for each of said "n virtual pipelines", which would only be possible if the virtual pipeline was considered a "virtual multithreaded processor" which actually is made up of multiple threads in a single pipeline (figs. 3 and 4). Applicant has described and shown in figs. 3 and 4, that each "virtual pipeline" (as described in the spec.) is made up of "k" stages IF, D,

E and W, however, each hardware (real) pipeline is made up of multiple virtual pipelines, i.e., IF1, D1, E1, W1 and IF2, D2, E2 and W2. In conclusion, it is unclear what a "virtual pipeline" refers to in light of the contradictions between the specification and claim limitations.

9. Further regarding claim 1, the limitation "reproducing said register set" and "adapting said reproduced register set" is indefinite. It is unclear if the step of "reproducing said register set" is intended to mean duplicating, e.g., making two register sets, or if it means using (or reusing) the register set from the "computer processor configuration" in the new, converted processor. Furthermore, if "reproducing" is intended to mean duplicating, then "said register set" is indefinite, because it is unclear to which register set it is referring to and how/why it stores machine states of all n virtual multithreaded pipelines. Further evidence that the register set is actually duplicated is found in claim 6, which claims, "Activating said register set that is associated with any of said selected n virtual multithreaded processors". Both interpretations have been examined below, the interpretation that the register set is actually duplicated for another thread is addressed in the 35 U.S.C. 103 Rejections, the interpretation that the register set is merely reused from the prior computer system and adapted for multiple threads is addressed in the 35 U.S.C. 102 Rejections.

10. Given the similarities between claim 1 and claim 13, the arguments as stated for the rejection of claim 1 also apply to claim 13.

11. As per claim 2, "said n virtual multithreaded pipelines" lacks antecedent basis. This further exemplifies the confusion of the limitations from claim 1. It is

unclear if "said n virtual multithreaded pipelines" is referring to the "n virtual multithreaded processors" or "n virtual pipelines" or if they are all the same entity.

12. Given the similarities between claim 2 and claim 14, the arguments as stated for the rejection of claim 2 also apply to claim 14.

New Claim Rejections - 35 USC § 102

13. The following is a quotation of the appropriate paragraphs of 35 U.S.C. 102 that form the basis for the rejections under this section made in this Office action:

> A person shall be entitled to a patent unless –
>
> (b) the invention was patented or described in a printed publication in this or a foreign country or in public use or on sale in this country, more than one year prior to the date of application for patent in the United States.

14. Claims 1-4 and 6-23 are rejected under 35 U.S.C. 102(e) as being anticipated by Lauterbach, U.S. P.G. Pub 2003/0046516.

15. Regarding claim 1, Lauterbach has taught a method of converting a computer processor configuration having a k-phased pipeline and a register set into n virtual multithreaded processors, where each of said n virtual multithreaded processors is compatible with said computer processor configuration, where n is a whole number equal to one and k is a whole number greater than zero, said method comprising the steps of: [Lauterbach teaches a computer processor configuration having a k-phased pipeline (k = 4) and a register set (Register File 106) [prior art, fig. 1] and converting this into n (n = 2) virtual multithreaded pipelined processors (fig. 2 and paragraphs 26 & 29). A virtual pipeline consists

of all the sub-stages of the pipeline during one clock cycle, which is equal to 8 sub-stages.]

a. Dividing each phase of said k-phased pipeline of said computer processor configuration into a plurality n of sub-phases [Figures 2 and 3 and Paragraph 32. K = 4 (F, D, E and W stages) and n = 2. Each of the k stages of the original pipeline from fig. 1 is split into a number of stages that is equal to the number of threads that are to be executed by the processor, which in the case of fig. 2, is n = 2.]; and

b. Creating n virtual pipelines within said k-phased pipeline, wherein each of said n virtual pipelines comprises n*k sub-phases: [Fig. 2, for each thread there is a sub-phase within the pipeline, i.e., there are 2 threads for the pipeline of fig. 2, therefore, each stage is divided into two sub-phases, see fig. 3. Each thread's pipeline is made up of one sub-stage from each stage, i.e., causing a thread's pipeline to be the same number of sub-stages as there are stages (k = 4). A virtual pipeline consists of all of the threads' pipelines at one particular clock cycle, thus creating n*k sub-phases. See fig. 2, Cycle 0 has one associated virtual pipeline made up of the two sub-stages of each of the Fetch, Decode, Execute and Write Back stages, and cycle 1 has a second similar virtual pipeline.]

c. Reproducing said register set of said computer processor configuration and adapting said reproduced register set to simultaneously store machine states of said n virtual multithreaded processors:

[Lauterbach teaches a register file 206 which has been adapted from the register file 106 of figure 1 to be able to handle multiple threads requests for operands, and inherently the storing of data in destination registers as well. Figure 2 shows that the register file holds data for both thread 0 and thread 1, therefore, there has inherently been the step of adapting the reproduced register set to store machine states of said n virtual multithreaded processors. (paragraphs 5 and 6, and paragraphs 26-28).]

16. Examiner notes that Lauterbach teaches a prior art implementation of a single-threaded, pipelined processor. This prior art processor's pipeline is shown in figure 1 and the converted version is shown in figure 2. The processor of figure 2 is a multithreaded, virtual pipelined processor as described above and in paragraphs 26-33. It has the same pipeline as the prior art processor, except for a few variations, including: 1) divided stages; 2) virtual pipelines; 3) reproduced register set [206 in figure 2, 106 in figure 1]; 4) an adapted register set which stores machine states for all the threads and virtual pipelines [item 206 in figure 2]. It is inherent that, since there are now divided stages, there was a step of dividing. It is inherent that, since there are now virtual pipelines, that there was a step of creating them. It is inherent that, since there is a single register set which operates the same way as in the prior art processor, the register set's functionality has been reproduced and therefore, said register set has been reproduced. Finally, it is inherent that since the reproduced register set now handles multiple virtual pipelines and threads, it has been adapted to do so.

17. Regarding claim 2, Lauterbach has taught a method according to claim 1 and further comprising executing at least one different thread within each one of said virtual multithreaded pipelines [Paragraph 32, each virtual multithreaded processor/pipeline has two threads concurrently executing within them. For instance, if there is a "thread 1" and a "thread 2" (as is the case of fig. 2), there is at least one different thread in each (thread 2 is different than thread 1 in each instance, or vice-versa)].

18. Regarding claim 3, Lauterbach has taught a method according to claim 2 wherein said executing step comprises executing any of said threads at an effective clock rate equal to the clock rate of said k-phased pipeline. [Since Lauterbach has sub-divided the stages into a number of sub-stages equal to the number of threads, the clock rate would have increased by a multiple of that number, thereby executing the threads at an effective rate equal to the non-divided processor. Paragraphs 29 and 32.]

19. Regarding claim 6, Lauterbach has taught a method according to claim 2 and further comprising: selecting any of said threads at a clock cycle; and activating at said clock cycle the register set that is associated with said selected thread. [Selecting a thread is the basic idea behind interleaved multithreading, as is disclosed by Lauterbach. Lauterbach teaches in figure 2 that in different cycles, the register set 206 is activated for a different selected thread, either thread 0, as in cycle 1 or thread 1, as in cycle 0.]

20. Regarding claim 7, Lauterbach has taught a method according to claim 1 wherein any of said steps are applied to a single-threaded processor

configuration [Figures 1 & 2, the prior art implementation is shown in figure 1 and is a single-threaded processor, and it is converted through the method taught by Lauterbach to be a virtual pipelined multithreaded processor.

21. Regarding claim 8, Lauterbach has taught a method according to claim 1 wherein any of said steps are applied to a multithreaded processor configuration. [Lauterbach, figure 2 and paragraph 32 teach that the multithreaded processor in fig. 2 can be altered to include more sub-stages, one for each thread.]

22. Regarding claim 9, Lauterbach has taught a method according to claim 1 wherein any of said steps are applied to said processor configuration a plurality of times for a plurality of different values of n, thereby creating a plurality of different processor configurations [Paragraph 32].

23. Regarding claim 10, Lauterbach has taught a method according to claim 1 wherein any of said steps are applied to said processor configuration a plurality of times for a plurality of different values of n [Figure 2 and Paragraph 32] until a target processor performance level is achieved. [Applying the steps a plurality of times would merely deepen the pipeline by several phases at a time, basically achieving the same goal as if applying the steps once with n being equal to the aggregate number of phases/threads, which is disclosed by Lauterbach. As for the performance level, Lauterbach's predefined target performance level is the number of threads desired to run on the processor, and the value of n is the number of threads, thus the number of times the stages are sub-divided.]

24. Regarding claim 11, Lauterbach has taught a method according to claim 1 wherein said dividing step comprises: selecting a predefined target processor

performance value; and selecting a value of n being in predefined association with said predefined target processor performance level. [Lauterbach's predefined target performance level is the number threads desired to run on the processor, and the value of n is the number of threads, thus the number of times the stages are sub-divided.]

25. Regarding claim 12, Lauterbach teaches the method according to claim 1, wherein said computer processor configuration is a synchronous logic block: [The prior art computer processor configuration (fig. 1) contains synchronous logic and therefore is a synchronous logic block. (See also paragraphs 4-5.)]

26. Given the similarities between claim 1 and claim 13, the arguments as stated for the rejection of claim 1 also apply to claim 13. The only difference between the claim limitations is the preamble, which in claim 1 calls for a method claim 13, which requires an apparatus, which essentially carries out the method of claim 1. However, an apparatus inherently carries out the method of converting the prior art processor of figure 1 to the multithreaded, virtual pipelined processor of figure 2.

27. Given the similarities between claim 2 and claim 14, the arguments as stated for the rejection of claim 2 also apply to claim 14.

28. Given the similarities between claim 3 and claim 15, the arguments as stated for the rejection of claim 3 also apply to claim 15.

29. Given the similarities between claim 4 and claim 16, the arguments as stated for the rejection of claim 4 also apply to claim 16.

30. Given the similarities between claim 6 and claim 17, the arguments as stated for the rejection of claim 6 also apply to claim 17.

31. Given the similarities between claim 7 and claim 18, the arguments as stated for the rejection of claim 7 also apply to claim 18.

32. Given the similarities between claim 8 and claim 19, the arguments as stated for the rejection of claim 8 also apply to claim 19.

33. Given the similarities between claim 9 and claim 20, the arguments as stated for the rejection of claim 9 also apply to claim 20.

34. Given the similarities between claim 10 and claim 21, the arguments as stated for the rejection of claim 10 also apply to claim 21.

35. Given the similarities between claim 11 and claim 22, the arguments as stated for the rejection of claim 11 also apply to claim 22.

36. Given the similarities between claim 12 and claim 23, the arguments as stated for the rejection of claim 12 also apply to claim 23.

New Claim Rejections - 35 USC § 103

37. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

38. Claim 4 is rejected under 35 U.S.C. 103(a) as being unpatentable over Lauterbach, U.S. P.G. Pub 2003/0046516.

39. Regarding claim 4, Lauterbach has taught a method according to claim 1 wherein said dividing step comprises: wherein each of said n sub-phases has a propagation delay of less than T/n [Paragraph 29, the total of one stage's sub-phases' propagation delays is equal to the associated stage of the prior art computer processor configuration of fig. 1.

40. Lauterbach fails to teach determining a minimum cycle time T=1/f for said computer processor configuration.

41. However, Examiner takes Official Notice that finding a minimum cycle time for pipeline stages in a processor is well known in the art, since it inherently maximizes the speed at which instructions are propagated through the pipeline and thus improves instruction processing speed as well.

42. It would have been obvious to one of ordinary skill in the art at the time the invention was made to minimized the cycle time for pipeline stages since Examiner takes Official Notice doing so will maximize the rate at which instructions/signals are propagated though the pipeline and therefore improve processing performance.

43. Claims 1-4 and 6-23 are rejected under 35 U.S.C. 103(a) as being unpatentable over Lauterbach, U.S. P.G. Pub 2003/0046516, in view of Culler, "Parallel Computer Architecture," herein referred to as Culler.

44. Regarding claim 1, Lauterbach has taught a method of converting a computer processor configuration having a k-phased pipeline and a register set into n virtual multithreaded processors, where each of said n virtual multithreaded processors is compatible with said computer processor configuration, where n is

a whole number equal to one and k is a whole number greater than zero, said method comprising the steps of: [Lauterbach teaches a computer processor configuration having a k-phased pipeline (k = 4) and a register set (Register File 106) [prior art, fig. 1] and converting this into n (n = 2) virtual multithreaded pipelined processors (fig. 2 and paragraphs 26 & 29). A virtual pipeline consists of all the sub-stages of the pipeline during one clock cycle, which is equal to 8 sub-stages.]

d. Dividing each phase of said k-phased pipeline of said computer processor configuration into a plurality n of sub-phases [Figures 2 and 3 and Paragraph 32. K = 4 (F, D, E and W stages) and n = 2. Each of the k stages of the original pipeline from fig. 1 is split into a number of stages that is equal to the number of threads that are to be executed by the processor, which in the case of fig. 2, is n = 2.]; and

e. Creating n virtual pipelines within said k-phased pipeline, wherein each of said n virtual pipelines comprises n*k sub-phases: [Fig. 2, for each thread there is a sub-phase within the pipeline, i.e., there are 2 threads for the pipeline of fig. 2, therefore, each stage is divided into two sub-phases, see fig. 3. Each thread's pipeline is made up of one sub-stage from each stage, i.e., causing a thread's pipeline to be the same number of sub-stages as there are stages (k = 4). A virtual pipeline consists of all of the threads' pipelines at one particular clock cycle, thus creating n*k sub-phases. See fig. 2, Cycle 0 has one associated virtual pipeline made up of the two sub-stages of each of the Fetch, Decode,

Execute and Write Back stages, and cycle 1 has a second similar virtual pipeline.]

45. However, Lauterbach fails to teach reproducing said register set of said computer processor configuration and adapting said reproduced register set to simultaneously store machine states of said n virtual multithreaded processors in the sense that the register set is actually duplicated, and there is one for each thread. (Lauterbach teaches one register file 206 which contains registers for each/all threads (fig. 2)).

46. However, Culler teaches an interleaved multithreading scheme that utilized replicated register sets, one for each of the threads, in order for the threads to function independently without register conflicts. Each of the register sets stores machine states (register values) for the associated thread.

47. Accordingly, it would have been obvious to one having ordinary skill in the art at the time the invention was made to utilize the teachings of Culler in the invention of Lauterbach and provide replicated register sets so that the independent threads run without encountering register conflicts between threads, since register conflicts degrade performance. The advantages of replicating the register sets as taught by Culler provides sufficient suggestion and motivation to one of ordinary skill in the art to follow the teachings of Culler and modify the invention of Lauterbach to allow each thread its own register set.

48. Regarding claim 2, Lauterbach, in view of Culler, has taught a method according to claim 1 and further comprising executing at least one different thread within each one of said virtual multithreaded pipelines [Paragraph 32,

each virtual multithreaded processor/pipeline has two threads concurrently executing within them. For instance, if there is a "thread 1" and a "thread 2" (as is the case of fig. 2), there is at least one different thread in each (thread 2 is different than thread 1 in each instance, or vice-versa)].

49. Regarding claim 3, Lauterbach, in view of Culler has taught a method according to claim 2 wherein said executing step comprises executing any of said threads at an effective clock rate equal to the clock rate of said k-phased pipeline. [Since Lauterbach has sub-divided the stages into a number of sub-stages equal to the number of threads, the clock rate would have increased by a multiple of that number, thereby executing the threads at an effective rate equal to the non-divided processor. Paragraphs 29 and 32.]

50. Regarding claim 4, Lauterbach has taught a method according to claim 1 wherein said dividing step comprises: wherein each of said n sub-phases has a propagation delay of less than T/n [Paragraph 29, the total of one stage's sub-phases' propagation delays is equal to the associated stage of the prior art computer processor configuration of fig. 1.

51. Lauterbach fails to teach determining a minimum cycle time T=1/f for said computer processor configuration.

52. However, Examiner takes Official Notice that finding a minimum cycle time for pipeline stages in a processor is well known in the art, since it inherently maximizes the speed at which instructions are propagated through the pipeline and thus improves instruction processing speed as well.

53. It would have been obvious to one of ordinary skill in the art at the time the invention was made to minimized the cycle time for pipeline stages since Examiner takes Official Notice doing so will maximize the rate at which instructions/signals are propagated though the pipeline and therefore improve processing performance.

54. Regarding claim 6, Lauterbach and Culler have taught a method according to claim 2 and further comprising: selecting any of said threads at a clock cycle; and activating at said clock cycle the register set that is associated with said selected thread. Selecting a thread is the basic idea behind interleaved multithreading, as disclosed by Lauterbach and Culler. As for the register sets, Culler discloses a multithreaded pipelined processor with replicated register sets, one for each thread (Culler: Page 905, The Basic Interleaved Scheme). In order for the registers of a given thread to be accessible when the corresponding thread is activated, the corresponding register set would also have been activated.

55. Regarding claim 7, Lauterbach, in view of Culler, has taught a method according to claim 1 wherein any of said steps are applied to a single-threaded processor configuration [Figures 1 & 2, the prior art implementation is shown in figure 1 and is a single-threaded processor, and it is converted through the method taught by Lauterbach to be a virtual pipelined multithreaded processor.

56. Regarding claim 8, Lauterbach, in view of Culler, has taught a method according to claim 1 wherein any of said steps are applied to a multithreaded processor configuration. [Lauterbach, figure 2 and paragraph 32 teach that the

multithreaded processor in fig. 2 can be altered to include more sub-stages, one for each thread.]

57. Regarding claim 9, Lauterbach, in view of Culler, has taught a method according to claim 1 wherein any of said steps are applied to said processor configuration a plurality of times for a plurality of different values of n, thereby creating a plurality of different processor configurations [Paragraph 32].

58. Regarding claim 10, Lauterbach, in view of Culler, has taught a method according to claim 1 wherein any of said steps are applied to said processor configuration a plurality of times for a plurality of different values of n [Figure 2 and Paragraph 32] until a target processor performance level is achieved. [Applying the steps a plurality of times would merely deepen the pipeline by several phases at a time, basically achieving the same goal as if applying the steps once with n being equal to the aggregate number of phases/threads, which is disclosed by Lauterbach. As for the performance level, Lauterbach's predefined target performance level is the number of threads desired to run on the processor, and the value of n is the number of threads, thus the number of times the stages are sub-divided.]

59. Regarding claim 11, Lauterbach, in view of Culler, has taught a method according to claim 1 wherein said dividing step comprises: selecting a predefined target processor performance value; and selecting a value of n being in predefined association with said predefined target processor performance level. [Lauterbach's predefined target performance level is the number threads desired

to run on the processor, and the value of n is the number of threads, thus the number of times the stages are sub-divided.]

60. Regarding claim 12, Lauterbach, in view of Culler, teaches the method according to claim 1, wherein said computer processor configuration is a synchronous logic block: [The prior art computer processor configuration (fig. 1) contains synchronous logic and therefore is a synchronous logic block. (See also paragraphs 4-5.)]

61. Given the similarities between claim 1 and claim 13, the arguments as stated for the rejection of claim 1 also apply to claim 13. The only difference between the claim limitations is the preamble, which in claim 1 calls for a method claim 13, which requires an apparatus, which essentially carries out the method of claim 1. However, an apparatus inherently carries out the method of converting the prior art processor of figure 1 to the multithreaded, virtual pipelined processor of figure 2.

62. Given the similarities between claim 2 and claim 14, the arguments as stated for the rejection of claim 2 also apply to claim 14.

63. Given the similarities between claim 3 and claim 15, the arguments as stated for the rejection of claim 3 also apply to claim 15.

64. Given the similarities between claim 4 and claim 16, the arguments as stated for the rejection of claim 4 also apply to claim 16.

65. Given the similarities between claim 6 and claim 17, the arguments as stated for the rejection of claim 6 also apply to claim 17.

66. Given the similarities between claim 7 and claim 18, the arguments as stated for the rejection of claim 7 also apply to claim 18.

67. Given the similarities between claim 8 and claim 19, the arguments as stated for the rejection of claim 8 also apply to claim 19.

68. Given the similarities between claim 9 and claim 20, the arguments as stated for the rejection of claim 9 also apply to claim 20.

69. Given the similarities between claim 10 and claim 21, the arguments as stated for the rejection of claim 10 also apply to claim 21.

70. Given the similarities between claim 11 and claim 22, the arguments as stated for the rejection of claim 11 also apply to claim 22.

71. Given the similarities between claim 12 and claim 23, the arguments as stated for the rejection of claim 12 also apply to claim 23.

Response to Arguments

1. Applicants arguments filed on 9/27/2005 have been fully considered but they are not persuasive.

2. Applicant argues the novelty/rejection of independent claims 1 and 12, however claims 1 and 13 are the only independent claims, so it will be assumed that the arguments for claim 12 were intended to be regarding claim 13.

"The process described by Lauterbach relates to a pipeline that is already configured to accept instructions from multiple independent threads of operation. The method divides each pipeline stage into two different stages, where on sub-stage can be processing an instruction from one thread while a second sub-stage is processing an instruction from a different thread. However, Lauterbach discloses nothing regarding a method or apparatus for converting a computer processor configuration into n virtual multithreaded processors while reproducing the register file to support storing and restoring of the multithreads in the processor."

"In contrast, the present invention discloses a method and apparatus for converting a computer processor configuration having one or more threads into n virtual multithreaded processors, where each of the n virtual processors is provided to execute the one or more threads and is further compatible to the original computer processor configuration both in instruction set and in execution time."

"Furthermore, the present invention describes the manner in which the execution phase is carried out by the reproduction of the register set and the implementation of selected logic, page 7 and 8, for combining the multithreading procedure and pipeline splitting within an existing and mature computer processor configuration."

3. These arguments are not found persuasive for the following reasons:

a. To clarify, applicant's attention is directed towards figures 1-3 and paragraphs 45 and 26-32. Figure 1 and paragraphs 4-5 and 26 describe the prior art pipelined computer ("mature computer processor configuration"). This prior art implementation is converted to be the multithreaded, virtual pipelined processor shown in figure 2. As described in paragraphs 26-29, figure 2 shows a pipeline almost identical to that of the one in figure 1, only varying in that the processor of figure 2 is not multithreaded using virtual pipelines and sub-phases within the original pipeline stages. Therefore, the processor of figure 1 has been converted to the processor of figure 2, and an instruction passes through the pipeline such that the total time spent in the sub-phases of a stage is equal to the time spent in one stage of the prior art processor of figure 1. Examiner notes that the method of conversion is not explicitly stated in the disclosure. However, figure 2 shows the result of the (inherent) method of conversion, since it shows the same pipeline of figure 1, however, it has been converted to become a multithreaded, virtual pipelined processor.

b. Regarding the conversion to reproduce "the register file to support storing and restoring of the multithreads in the processor," Examiner agrees and acknowledges that Lauterbach does not teach this aspect. Lauterbach teaches one register file 206, which handles both threads' requests for data operands, and therefore teaches adapting the register file to handle machine states for each thread/virtual multithreaded processors. However, there is no reproduction (e.g., duplication) of the register file/set for each thread taught, Lauterbach is silent on how the register file 206 is used or what it is made up.

c. However, as addressed in the 35 U.S.C. 103 Rejection above, Culler does teach this aspect of reproducing register sets/files, one for each thread. Culler also teaches the advantages of this, and they are stated in the above Rejection of claim 1 as well.

4. Applicant argues the novelty/rejection of dependent claims 5 and 6, however, claim 5 has been cancelled and its limitations moved to independent claims 1 and 13.

"Culler describes an interleaved multithreading scheme, whereas the active contexts are supported by replicating the register set and other critical state times, see Culler; The Interleaved Scheme, page 905."

"However, Culler suggests nothing about introducing a combination of multithreading and pipeline splitting when converting a computer processor configuration into a virtual multithreaded processor, where the virtual multithreaded processor is compatible to the original computer processor configuration in both execution behavior and execution time."

5. These arguments are not found persuasive for the following reasons:

d. Culler teaches a finalized multithreaded scheme, which includes a register file to hold machine states for each active thread in a

multithreaded processor. Inherently, when designing this system, the step of reproducing register files for each thread had to occur, since they exist in the system. Also, inherently, when designing the system, the reproduced register set had to be adapted to simultaneously store machine states of the threads, since the final product teaches a reproduced register set adapted to simultaneously store machine states, the step of adapting must have occurred.

e. Furthermore, no assertion has been made by the Examiner that Culler teaches introducing a combination of multithreading and pipeline splitting when converting a computer processor configuration into a virtual multithreaded processor. Examiner has, however, asserted that Lauterbach teaches these limitations.

6. The arguments regarding the Hennessy reference (page 12 of Remarks) are found moot in view of the new rejections above, which do not cite Hennessy.

7. Lastly, the concluding arguments by Applicant regarding the patentability of all the claims are found unpersuasive for the reasons given above in the response to arguments and the new rejections.

Conclusion

8. **THIS ACTION IS MADE FINAL.** Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

The following is text cited from 37 CFR 1.111(c): In amending in reply to a rejection of claims in an application or patent under reexamination, the applicant

or patent owner must clearly point out the patentable novelty which he or she thinks the claims present in view of the state of the art disclosed by the references cited or the objections made. The applicant or patent owner must also show how the amendments avoid such references or objections.

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the mailing date of this final action.

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Kevin P Rizzuto whose telephone number is (571) 272-4174. The examiner can normally be reached on M-F, 8-4:30.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Eddie Chan can be reached on (571) 272-4162. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

KPR

EDDIE CHAN
SUPERVISORY PATENT EXAMINER
TECHNOLOGY CENTER 2100